CLAIMS

We claim:

1. A method for avoiding cache pollution in a microprocessor system comprising:

determining a dependency state condition between a first load instruction and a second load instruction;

storing said dependency state condition in a dynamic memory dependency table;

determining a miss state condition in a cache for said second load instruction;

decoding said miss state condition for said second load instruction to a decoded miss state condition ID;

logically combining said dependency state condition and said decoded miss state condition ID; and

quashing execution of said second load instruction if said logically combining results in a true condition.

2. The method of Claim 1 wherein said logically combining comprises logically ANDing.

3. The method of Claim 2 further comprising rescheduling said second load instruction for execution if said logically combining results in a true condition.

4. The method of Claim 1 wherein said dynamic memory dependency table comprises random access memory cells.

5. The method of Claim 1 wherein said logically combining said dependency state and said decoded miss state condition ID employs a logical combining gate in a gate array.

6. A microprocessor system for avoiding cache pollution comprising:

a dependency checker for determining a dependency state condition between a first load instruction and a second load instruction;

an instruction scheduler coupled to said dependency checker, said instruction scheduler having an instruction buffer storing said first load instruction and said second load instruction, said instruction scheduler scheduling execution of said first load instruction and said second load instruction;

an execution unit coupled to said instruction scheduler;

a cache system having a cache memory and coupled to said execution unit, said cache system having a cache miss state condition output for a cache miss state condition in said cache memory for said first load instruction;

a cache pollution avoidance unit coupled to said cache system and coupled to said dependency checker said cache pollution avoidance unit comprising:

a dynamic memory dependency table storing said dependency state condition between said first load instruction and said second load instruction;

a cache miss ID decode unit having a cache miss ID decode input coupled to said cache miss state condition output of said cache system and having a cache miss ID decode output;

a logical combining gate having a gate first input coupled to said dependency state condition between said first load instruction and said second load instruction stored in said dynamic memory dependency table, said logical combining gate having a gate second input coupled to said cache miss ID decode output of said cache miss ID decode unit, said logical combining gate having a gate output; and

wherein said instruction scheduler squashes execution by said execution unit of said second load instruction stored in said instruction buffer whenever said output of said logical combining gate is true.

7. The microprocessor system for avoiding cache pollution of Claim 6 wherein said logical combining gate comprises a logically ANDing gate.

8. The microprocessor system for avoiding cache pollution of Claim 6 wherein said scheduling unit re-schedule execution of said second load instruction whenever said output of said logical combining gate is true.

9. The microprocessor system for avoiding cache pollution of Claim 6 wherein said dynamic memory dependency table comprises random access memory cells.

10. The microprocessor system for avoiding cache pollution of Claim 6 wherein said logically combining gate is a member of a gate array.